

## WHAT IS CLAIMED IS:

1           1.     A packet transformation module for editing multi-protocol streaming  
 2 data packets, comprising:  
 3                 an instruction memory coupled to receive search words identifying a  
 4 packet type for the packet, wherein the instruction memory outputs appropriate  
 5 instructions based on the packet type as indexed by the search words;  
 6                 a packet memory coupled to receive one or more portions of the  
 7 packet subject to editing, wherein each of the packet portions is stored in a  
 8 respective memory segment of the packet memory;  
 9                 a valid bit array having a plurality of memory validity fields associated  
 10 with respective memory segments, wherein the state of each of the memory validity  
 11 fields establishes whether the packet portion in the respective memory segment is  
 12 incorporated into a resulting packet portion; and  
 13                 a processing module coupled to receive the instructions from the  
 14 instruction memory, and to edit the packet portions in accordance with the  
 15 instructions.

1           2.     The packet transformation module as in Claim 1, further comprising a  
 2 priority encoder coupled to the valid bit array to enable output of each of the packet  
 3 portions associated with asserted ones of the memory validity fields.

1           3.     The packet transformation module as in Claim 2, further comprising an  
 2 output controller coupled to the priority encoder to output each of the packet  
 3 portions enabled for output by the priority encoder.

1           4.     The packet transformation module as in Claim 3, further comprising an  
 2 overflow memory to receive and temporarily store second packet portions not  
 3 subject to editing.

1           5.     The packet transformation module as in Claim 4, further comprising a  
 2 multiplexer coupled to the output controller and the overflow memory to selectively

3 enable transmission of the resulting packet portion from the output controller  
4 followed by the second packet portions from the overflow buffer.

1 6. The packet transformation module as in Claim 1, wherein the packet  
2 portions are separated by one or more blank memory segments to facilitate insertion  
3 of additional information between existing packet portions stored in the packet  
4 memory.

1 7. The packet transformation module as in Claim 6, wherein the resulting  
2 packet portion is read out of the packet memory in an order that the packet portions  
3 are stored in the packet memory.

1 8. The packet transformation module as in Claim 1, wherein the packet  
2 memory is a dual-port packet memory, and wherein the dual-port packet memory  
3 concurrently receives two portions of the packet that are subject to editing.

1 9. The packet transformation module as in Claim 1, further comprising a  
2 macro sequencer coupled to the packet memory to effect post-processing  
3 modifications to the packet portions in the packet memory.

1 10. The packet transformation module as in Claim 9, wherein the post-  
2 processing modifications comprise at least one of updating a checksum, a time-to-  
3 live field, a packet conformance marking, and a packet length.

1 11. A method for editing packets of a packet stream received at a network  
2 node, comprising:  
3 storing one or more segments of a packet in partitionable memory  
4 segments of a modification memory;  
5 eliciting one or more editing instructions from an instruction memory,  
6 wherein the particular editing instructions elicited is based at least in part on  
7 characteristics of the packet;

8                   modifying at least one packet segment stored in the modification  
9   memory as directed by the editing instructions;  
10                  associating validity tags with each of the memory segments to indicate  
11   whether or not their corresponding packet segments will be incorporated into a  
12   resulting modified packet; and  
13                  creating the resulting modified packet by assembling the packet  
14   segments associated with the validity tags indicating incorporation into the resulting  
15   modified packet.

1           12.   The method of Claim 11, further comprising receiving search words  
2   corresponding to the characteristics of the packet, wherein the search words are  
3   used to index the instruction memory and elicit appropriate editing instructions.

1           13.   The method of Claim 11, wherein modifying at least one packet  
2   segment of the modification memory includes changing, adding, or deleting one or  
3   more packet segments in the modification memory.

1           14.   The method of Claim 11, wherein modifying at least one packet  
2   segment of the modification memory comprises processing the editing instructions  
3   and modifying contents of the modification memory pursuant to the instructions.

1           15.   The method of Claim 11, wherein modifying at least one packet  
2   segment of the modification memory comprises canceling at least one packet  
3   segment by associating an invalid tag with each of the memory segments  
4   associated with the canceled packet segment, wherein the invalid tag indicates that  
5   the packet segment will not be incorporated into the resulting modified packet.

1           16.   The method of Claim 11, wherein storing the packet segments  
2   comprises reserving memory space between the stored packet segments.

1           17. The method of Claim 16, wherein modifying at least a portion of the  
2 modification memory comprises inserting new packet information in the reserved  
3 memory space.

1           18. The method of Claim 11, further comprising interleaving available  
2 memory between the parsed portions of the packet in the modification memory, in  
3 order to allow insertion of new packet segments.

1           19. The method of Claim 11, wherein associating validity tags comprises  
2 setting a field of one or more bits to a first state when the packet segment is to be  
3 included in the modified packet, and setting the field to a second state when the  
4 packet segment is not to be included in the modified packet.

1           20. The method of Claim 11, wherein storing the packet segments in the  
2 modification memory comprises storing the packet segments in a predetermined  
3 arrangement in the modification memory.

1           21. The method of Claim 20, wherein creating the modified packet  
2 comprises outputting the packet segments in order of the predetermined  
3 arrangement.

1           22. The method of Claim 11, wherein creating the modified packet  
2 comprises assembling the packet segments in ascending address order of the  
3 modification memory.

1           23. The method of Claim 11, wherein creating the resulting modified  
2 packet comprises assembling the packet segments in a predetermined order.

1           24. The method of Claim 11, further comprising determining the  
2 characteristics of the packet, including a transmission protocol associated with the  
3 packet.

1           25.    The method of Claim 24, wherein determining the characteristics of the  
2 packet comprises parsing header information of the packet to identify the  
3 transmission protocol associated with the packet.

1           26.    The method of Claim 11, wherein storing packet segments comprises  
2 storing parsed header information corresponding to embedded network layer  
3 headers within the packet.

1           27.    The method of Claim 11, further comprising effecting policing  
2 modifications through direct manipulation of the packet segment in the modification  
3 memory after the packet segment has been modified pursuant to the editing  
4 instructions.

1           28.    The method of Claim 11, further comprising serially receiving the  
2 packets from the packet stream.

1           29.    The method of Claim 11, further comprising transmitting the resulting  
2 modified packet from the network node.

1           30.    The method of Claim 11, further comprising separating packet  
2 segments subject to modification from packet segments not subject to modification.

1           31.    The method of Claim 30, further comprising storing the packet  
2 segments not subject to modification in an overflow buffer, wherein no modifications  
3 of the packet segments in the overflow buffer are effected.

1           32.    The method of Claim 30, wherein the packet segments subject to  
2 modification include packet header information.

1           33.    The method of Claim 30, wherein the packet segments not subject to  
2 modification include packet payload information.

1           34.    The method of Claim 11, further comprising:  
2                   storing segments of the packet which are not to be modified in an  
3 overflow memory;  
4                   performing the modifications of the packet segments stored in the  
5 modification memory; and  
6                   reassembling the packet to create the resulting modified packet.

1           35.    An ingress processing system for editing packets of a data stream,  
2 comprising:

3                   (A)    a packet parser to parse each packet and generate  
4 resulting search words based on a packet protocol; and

5                   (B)    an editor for editing the packets, comprising:

6                       (i)    an instruction memory coupled to the packet parser to  
7 receive the search words, wherein the instruction memory outputs targeted  
8 instructions based on the packet protocol as indexed by the search words;

9                       (ii)   a packet memory coupled to receive one or more  
10 portions of the packet subject to editing, wherein each of the packet portions  
11 is stored in a respective memory segment of the packet memory;

12                       (iii)   a valid bit array having a plurality of memory validity  
13 fields associated with respective memory segments, wherein the state of  
14 each of the memory validity fields establishes whether the packet portion in  
15 the respective memory segment is incorporated into a resulting edited packet;  
16 and

17                       (iv)   a processing module coupled to receive the instructions from  
18 the instruction memory, and to edit the packet portions in accordance with the  
19 instructions.

1           36.    A network for transferring information, comprising:

2                   (A) a source node to dispatch information onto the network;

3                   (B) a destination node to receive the information dispatched by the  
4 source node;

5 (C) at least one intermediary node coupled along a transmission path  
6 between the source node and the destination node, wherein the intermediary node  
7 comprises:

8 (i) a packet parser to parse each packet and generate  
9 resulting search words based on a packet protocol; and

10 (ii) an editor for editing the packets, comprising:

11 (a) an instruction memory coupled to the packet  
12 parser to receive the search words, wherein the instruction memory  
13 outputs targeted instructions based on the packet protocol as indexed  
14 by the search words;

15 (b) a packet memory coupled to receive one or more  
16 portions of the packet subject to editing, wherein each of the packet  
17 portions is stored in a respective memory segment of the packet  
18 memory;

19 (c) a valid bit array having a plurality of memory  
20 validity fields associated with respective memory segments, wherein  
21 the state of each of the memory validity fields establishes whether the  
22 packet portion in the respective memory segment is incorporated into  
23 a resulting edited packet; and

24 (d) a processing module coupled to receive the instructions from  
25 the instruction memory, and to edit the packet portions in accordance with the  
26 instructions.

1 37. The network of Claim 36, wherein the intermediary node further  
2 comprises a policing engine coupled to the editor to provide measured packet  
3 conformance parameters to the editor to carry out post-processing modifications.

1 38. The network of Claim 36, further comprising a switch fabric to direct  
2 the resulting edited packet to a subsequent node or the destination node.

1           39.    A computer-readable medium having computer-executable instructions  
2   for editing multi-protocol streaming data packets, the computer-executable  
3   instructions performing steps comprising:

4                   storing one or more segments of a packet in partitionable memory  
5   segments of a modification memory;

6                   eliciting one or more editing instructions from an instruction memory,  
7   wherein the particular editing instructions elicited is based at least in part on  
8   characteristics of the packet;

9                   modifying at least one packet segment stored in the modification  
10   memory as directed by the editing instructions;

11                   associating validity tags with each of the memory segments to indicate  
12   whether or not their corresponding packet segments will be incorporated into a  
13   resulting modified packet; and

14                   creating the resulting modified packet by assembling the packet  
15   segments associated with the validity tags indicating incorporation into the resulting  
16   modified packet.

1           40.    A packet transformation module for editing multi-protocol streaming  
2   data packets, comprising:

3                   means for storing one or more segments of a packet in partitionable  
4   memory segments of a modification memory;

5                   means for eliciting one or more editing instructions from an instruction  
6   memory, wherein the particular editing instructions elicited is based at least in part  
7   on characteristics of the packet;

8                   means for modifying at least one packet segment stored in the  
9   modification memory as directed by the editing instructions;

10                   means for associating validity tags with each of the memory segments  
11   to indicate whether or not their corresponding packet segments will be incorporated  
12   into a resulting modified packet; and



- 13 means for creating the resulting modified packet by assembling the  
14 packet segments associated with the validity tags indicating incorporation into the  
15 resulting modified packet.

continued